ARL-TR-7443 ● Sep 2015

ARL
US Army Research Laboratory

An Integrated Assessment of Progress in Robotic Perception and Semantic Navigation

by Craig Lennon, Barry Bodt, Marshal Childers, Jean Oh, Arne Suppe, Luis Navarro-Serment, Robert Dean, Terrence Keegan, Chip Diberardino, Menglong Zhu, and Sandong Park

Approved for public release; distribution is unlimited.

NOTICES

Disclaimers

The findings in this report are not to be construed as an official Department of the Army position unless so designated by other authorized documents.

Citation of manufacturer's or trade names does not constitute an official endorsement or approval of the use thereof.

Destroy this report when it is no longer needed. Do not return it to the originator.

ARL-TR-7443 ● SEP 2015

US Army Research Laboratory

An Integrated Assessment of Progress in Robotic Perception and Semantic Navigation

by Craig Lennon, Barry Bodt, and Marshal Childers
Vehicle Technology Directorate, ARL

Jean Oh, Arne Suppe, and Luis Navarro-Serment
National Robotics Engineering Center, Pittsburgh, PA

Robert Dean, Terrence Keegan, and Chip Diberardino
General Dynamics Land Systems, Westminster, MD

Menglong Zhu and Sandong Park
University of Pennsylvania, Philadelphia, PA

REPORT DOCUMENTATION PAGE	*Form Approved* *OMB No. 0704-0188*

1. REPORT DATE *(DD-MM-YYYY)* September 2105	**2. REPORT TYPE** Final	**3. DATES COVERED (From - To)** 1–31 October 2014

4. TITLE AND SUBTITLE An Integrated Assessment of Progress in Robotic Perception and Semantic Navigation	**5a. CONTRACT NUMBER**
	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S) Craig Lennon, Barry Bodt, Marshal Childers, Jean Oh, Arne Suppe, Luis Navarro-Serment, Robert Dean, Terrence Keegan, Chip Diberardino, Menglong Zhu, and Sandong Park	**5d. PROJECT NUMBER**
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) US Army Research Laboratory ATTN: RDRL-VTA Aberdeen Proving Ground, MD 5066	**8. PERFORMING ORGANIZATION REPORT NUMBER** ARL-TR-7443
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	**10. SPONSOR/MONITOR'S ACRONYM(S)**
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

The US Army Research Laboratory's Robotics Collaborative Technology Alliance (RCTA) is a program intended to change robots from tools that Soldiers use into teammates with which Soldiers can work. This requires the integration of fundamental and applied research in perception, artificial intelligence, and human-robot interaction. In October 2014, the RCTA assessed progress toward integrating this research. This assessment was designed to evaluate the robot's performance when it used new capabilities to perform selected aspects of a mission. The assessed capabilities included the ability of the robot to navigate semantically outdoors with respect to structures and landmarks, identify doors in the facades of buildings, and identify and track persons emerging from those doors. This report presents assessment results of each of these capabilities, evaluated separately but as a component of an integrated system.

15. SUBJECT TERMS

unmanned ground system, autonomous systems, semantic navigation, object detection, semantic perception, pedestrian tracking

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON** Craig Lennon
a. REPORT Unclassified	**b. ABSTRACT** Unclassified	**c. THIS PAGE** Unclassified	UU	48	**19b. TELEPHONE NUMBER (Include area code)** 410-278-9886

Standard Form 298 (Rev. 8/98)
Prescribed by ANSI Std. Z39.18

Contents

List of Figures

List of Tables

INTENTIONALLY LEFT BLANK.

1. Introduction

The US Army Research Laboratory's Robotics Collaborative Technology Alliance (RCTA) is an alliance of robotics research institutions working together to transform robots from tools that Soldiers use into teammates with which Soldiers can work.[1] Currently fielded ground robots require a dedicated operator to control them during their operation and perhaps additional Soldiers to provide security for the operator. Thus, while serving as useful tools for performing potentially dangerous tasks, they are an asset that requires a lot of attention. To serve as a true team member, the robot should have an understanding of its environment, the capability to perform its required tasks with only occasional guidance, and the ability to recognize when it needs assistance. As a guiding scenario, the RCTA has been using a "screen-the-back" scenario, in which a robot is told to go behind a building and watch the back door to see if any people exit through it. The robot should be able to receive the command verbally or in text and carry out the mission without an a priori map, recognizing buildings and other relevant landmarks as needed. It should notify its team members as to its progress and request clarification when uncertain as to how to interpret a command.

To test progress toward this goal, the RCTA capstone assessment evaluated how technologies developed by different members of the alliance functioned when integrated onto one system. The screen-the-back scenario begins when the robot receives instructions in a structured language. These instructions give positions to which the robot should navigate and objects to be used as landmarks. If the robot successfully navigates to the correct position, it will detect and orient toward a door on the building, subsequently detecting and tracking pedestrians exiting through the door. If it is not successful, a human-robot interaction (HRI) interface is available to get the robot back on track. Here, we present results from the experiments testing semantic navigation and perception, door detection, and pedestrian detection and tracking. These tests evaluated each component separately but as a component of the integrated system. Component testing of HRI was presented in Hill et al.[2]

Semantic navigation is navigation using semantically defined goals and landmarks, which are identified during the run by semantic perception. For example, one command given to the robot was "stay to the right of the building; navigate to the front of a traffic barrel behind the building". To be fully successful, the robot was expected to identify the building correctly and move around the right side of the building to within 3 m of a traffic barrel placed behind the building. The assessment of door and pedestrian detection began with the robot placed near a building having at least 2 doors. To be fully successful at door detection, the robot was expected to detect all doors (within view of its position) on the target building. Success at pedestrian detection required the detection and tracking of 1–3 pedestrians, who emerged from 1 or 2 doors, and walked away from the doors along various paths.

In the next section, we describe the integrated system and the technologies it used. Section 3 describes the methodology of the assessment, section 4 presents the assessment results, and the last section contains our conclusions.

2. The Integrated System

In this section, we describe the equipment and technology that played a key role in this assessment. We start with the robotic platform, including the sensors and computational power that support the intelligence architecture. Next is the perceptual system, which identifies objects in the world, and the world model, which is the robot’s internal representation of these objects. Then we describe aspects of the intelligence architecture in action: navigation, search, and observation. Navigation receives a navigation command and produces a plan for satisfying the requirements of the command. Search detects doors and orients the robot toward them. Observation detects and tracks pedestrians. A mission is decomposed into a sequence of actions (e.g., navigate, search, observe), where each action has its own goal. This goal is generally the precondition of the next action in the mission plan. In the screening mission, the robot first needs to navigate to the target area of reconnaissance, and when the robot has reached the target area, the search for doors is triggered. The completion of the search action is the precondition for the observe action. This chain of actions constitutes the mission. A mission planner monitors the chain, directing the next action to begin when the previous one has achieved its goal.

2.1 Robotic Platform

The robot used in the integrated assessment is a Clearpath Husky, equipped with the General Dynamics XR 3D LADAR sensor, Bumblebee stereo camera, and Adonis camera as shown in Fig. 1. The XR LADAR sensor is mounted 0.7 m above ground, which creates a dead zone around the robot of an approximately 4-m radius. A Hokuyo UTM-30LX scanning laser sensor is installed at 0.25 m for obstacle detection in the dead zone. Within the body of the robot are 4 Mac Mini machines, each with 2.3-GHz quad-core processors and 8-GB memory. During the testing, a laptop was tethered to the robot to act as a substitute for HRI modules, which were not present for the component tests. The computers run software modules from researchers at different institutions. These different software modules are integrated through the RFRAME framework developed at General Dynamics. RFRAME is a transport agnostic middleware, supporting multiple simultaneous protocols, (e.g., Joint Architecture for Unmanned Systems, Robot Operating System, and Neutral Message Language). By abstracting and optimizing differences between environments, RFRAME allows researchers to work in their preferred software environment but as part of an integrated system.

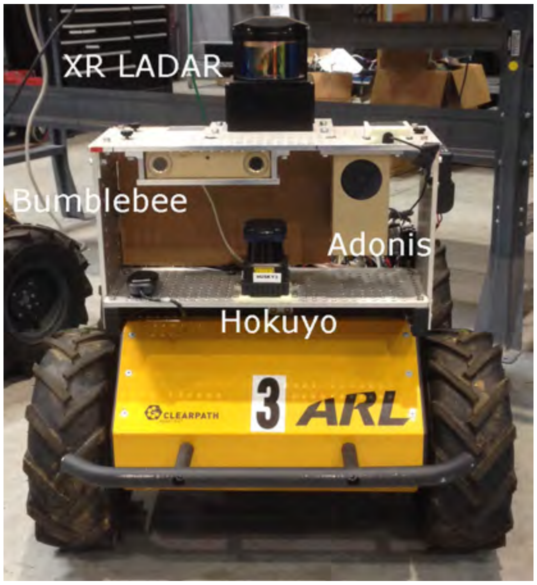


Fig. 1 Robot used for the assessment

2.2 Perception

The classification of terrain and detection of objects required different perceptual algorithms. First, a semantic classifier is used to classify regions of camera images.[3] Each pixel within the 2-dimensional (2-D) image is labeled as being one of several types: building, grass, tree, sky, concrete, asphalt, gravel, car, traffic barrel, fire hydrant, or unknown. Figure 2 shows an example of an outdoor scene with pixels colored by label. This approach to classification was shown by Lennon et al.[4] to perform well on outdoor terrain classification, although at the time of the study, the classification did not include the categories of traffic barrel, fire hydrant, or car. Once labeled, the 2-D images are fused with 3-dimensional (3-D) LADAR data to create colorized, semantically labeled 3-D point clouds. The points comprising the object may well have different labels, and the object label is chosen using a Bayesian approach.[5] Figure 2 shows a traffic barrel and a fire hydrant, with colored labels for the points. Although some points on the fire hydrant and traffic barrel are labeled as being of class "car" or "unknown", the correct objects are still detected. As the classification of whole scenes is the primary purpose of the semantic classifier, every pixel within each image is given some type of label, with the pixel labels chosen to give the best overall result, even at the expense of mislabeling small subregions. As objects tend to be small subregions when viewed from a distance, the semantic classifier thus sacrifices object detection capabilities for better overall scene recognition. Consequently, during the assessment, the semantic classifier was complemented by an object detector, which examined subregions of each image to detect specific objects: traffic barrels, fire hydrants, cars, and gas pumps.

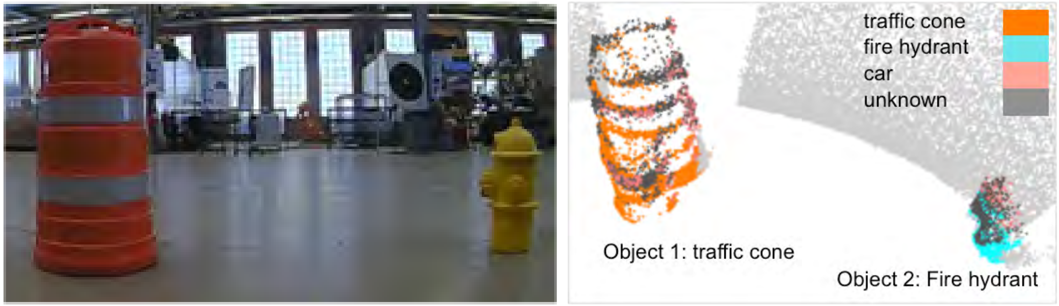
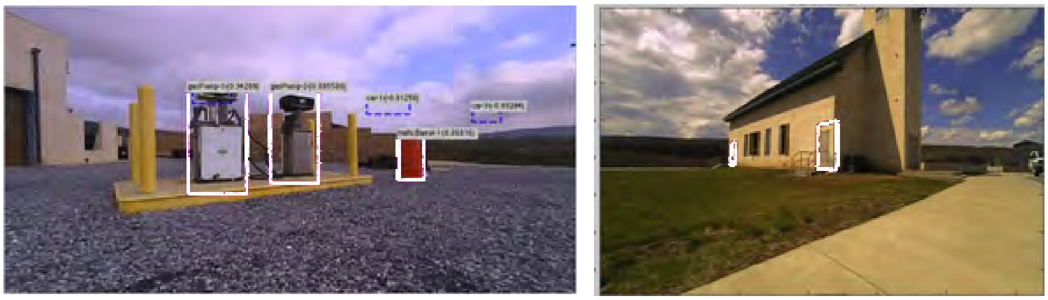


Fig. 2 An example of semantic object detection: actual objects (left), 3-D point cloud images (right)

This object detector used an active deformable part model (ADPM) method.[6] With deformable parts models, objects are recognized by their parts, which may be deformed into different configurations. Although classification by such methods can be slow, the implementation in Zhu et al.[6] allowed the classifier to run in real time on the robot. This detector was used for detecting traffic barrels, fire hydrants, gas pumps, and cars. Fig. 3a shows assessment results for ADPM object detection for finding gas pumps and a traffic barrel, and Fig. 3b shows doors detected by DPM and RANdom SAmple Consensus (RANSAC).



(a) (b)

Fig. 3 An example of ADPM object detection

A different deformable part model (DPM) was used for detecting doors. In this instance, the search was speeded up by the knowledge that a door can only be located on the vertical surface of a building. Thus, the door detection algorithm first detects façades using the RANSAC algorithm to narrow down the search area, and this allows the model to run in real time on the system. An example of the results of door detection is shown in Fig. 3. This real-time perception allows the robot to maintain a current model of the world around it. Semantic objects detected through these perception approaches are added to the robot's world model, and are updated as the robot's viewpoint changes over time. All mission commands and planning are interpreted according to the robot's model of the world, which we now describe.

2.3 Common World Model

The intelligence architecture is built around a common world model (CWM).[7] Traditional robot architectures define the world model as an abstract notion, which contains the various data and intermediate representations. The CWM defines and instantiates the data model for the intelligence architecture with a data-centric approach, providing common, centralized, and intelligent data store services.

The world model combines data that is metric (e.g., sensor data and aggregates), and semantic (e.g., class descriptions and instances), with the robot's self-knowledge (e.g., position, mission status, and goal). The world model is an intelligent data store, and not just a database. Internally, the world model knows how the various data sources interrelate, and, when appropriate, propagates changes between the metric, semantic, and self-levels. At the metric level, CWM efficiently represents and updates sensor data taken from a robot's environment in a multihypothesis, multiresolution map. At the semantic level, objects represent symbolic information, enabling the abstract reasoning needed for intelligent behavior. Here, CWM maintains semantic information from perception modules and provides methods for client modules to search for semantic objects that are relevant to a specific mission. For example, a point cloud is stored at the metric level, and classified as a wall. The label of wall is stored at the semantic level. A bounding box is established at the metric level, and adjacency to other walls and membership as part of a building is established and stored at the semantic level. There is a method for the navigate action to find this building when a navigate command is given with respect to a building, as well as methods for finding other relevant objects that are recorded in similar metric-semantic terms. Finally, self-information contains data relative to the robot itself. Tracking self-knowledge, such as current capability, component status, and task execution states, enables the robot to reason, and adapt its performance.

2.4 Navigation

Navigation begins with a command issued to the robot through an HRI interface. This command, called a tactical behavior specification (TBS), is in a structured language that is used for communication among software modules within the intelligence architecture. The TBS language supports a rich set of constraints that leverages spatial relationships among objects in an environment. As an example, consider the command "stay left of the building; navigate to a traffic barrel that is behind the building". The robot searches the world model for a building in front of it, predicts parts of the building it cannot observe, predicts a position for the traffic barrel behind the building, and plans a path to that goal.

In Fig. 4, the left image show the robot's camera view with some semantic labels. The right side show the robot's model of the world, which includes perceived objects, such as the front walls and the traffic barrel, as well as predicted objects, like the rest of the building and the predicted traffic barrel behind the building. In this example, the command includes 2 landmarks, a building and traffic barrel, but the robot's current world model contains only a set of walls and a predicted building. This inconsistency causes low grounding confidence, which, in turn, enables geometric spatial reasoning. Based on the context in the command, a traffic barrel must be behind the building, so an object is hypothesized behind the building. Now, the world model includes a building and a traffic barrel, both predicted. After symbol grounding is done with sufficiently high confidence, the robot computes a navigation cost map that best satisfies the action constraint to "stay left of the building" and plans a path accordingly. Technical details of the navigate action can be found in Oh et al.[5] When the navigate action is completed, the search action begins.

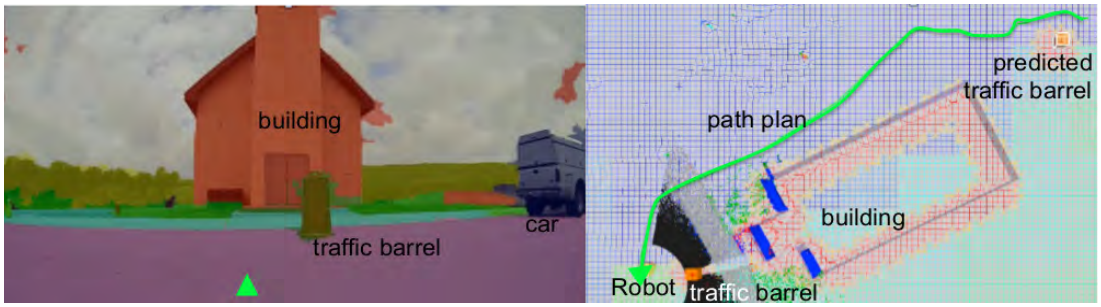


Fig. 4 The robot's view and predictions for a navigation TBS

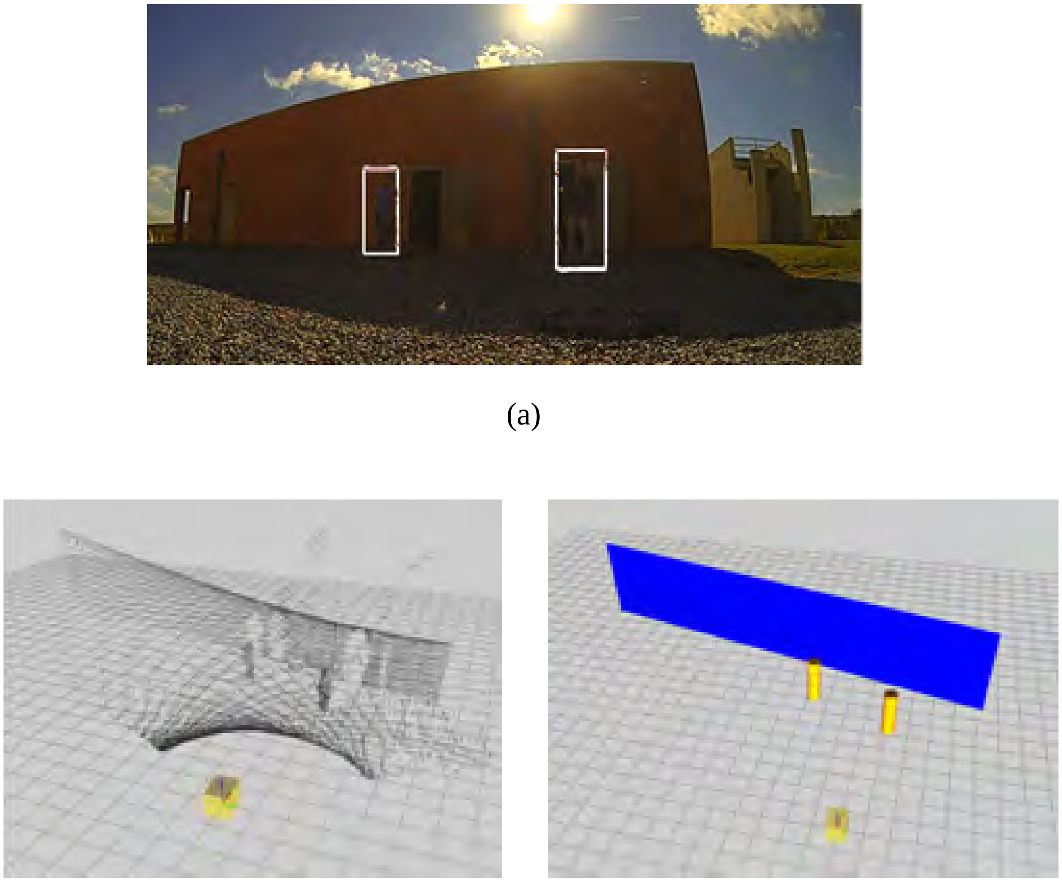
2.5 Search

The search action positions and orients the robot, relative to an object of interest to the human teammate. For example, with the command to "screen the back of the building", the detected building in the world model is the goal and the robot would reorient towards the center of the building to complete the navigate action. Once this orientation was achieved, the mission planner directs the search action to begin, and provides the type of object to search for (a door, in this assessment). The door detection algorithm is always running as part of the perception system, so doors in the scene might already be registered in the world model. In case they are not, the action provides a fixed amount of time for the door detection algorithm to report new detections. After this time expires, the search action will report the number of doors that it found within a configurable field of view. The intention is to have the human teammate choose an object from among the multiple options that would be displayed in the HRI interface (i.e., all doors found within the allotted time). Once the human teammate selects an object, the robot would then reorient toward that object to begin the observe action. This interaction was not tested as part of these preliminary experiments. Instead, the robot was programmed to orient toward the closest door to its current heading vector. Once this orientation was complete, the search action sent a message to the mission planner, and the mission planner directed the observe action to begin.

2.6 Observe

The observe action registers pedestrian detections and reports them to the world model. This action assumes that a previous action has positioned and oriented the robot relative to the object that is being observed. When the mission planner directs the observe action to start, the action begins listening to the output from the pedestrian detector that is already sending pedestrian detection messages. Pedestrian detection messages contain pixel locations for a box that encapsulates the individual parts of the detected person[8] for every detected person in the source image. The mapped LADAR pixels within each pedestrian detection box are segregated from the LADAR pixel cluster, and clustered using Euclidean Cluster Extraction.[9] If any of the resultant clusters fit a geometric heuristics check, and if there are no previous pedestrian tracks of the same shape close by, a new "person" object is added to the world model. Otherwise, if there is a track nearby that matches in shape, that track is updated. For this assessment, the robot continued to observe in this state until the system was shut down.

In the observe action example shown in Fig. 5, 2 people exited from the middle and right doors on the back of a building and stood stationary for approximately 5 s, allowing the pedestrian detection algorithm to publish detection boxes and correlate LADAR points in 3-D. They then walked adjacent to the back of the building until they were out of the LADAR's field of view. As shown in Fig. 5a, the lighting during this portion of the assessment provided challenges to the pedestrian detection. Figures 5b and 5c show the pedestrians in the world model as point clouds on the metric level (5b) and as semantic objects (5c). The sequential execution of the navigate, search, and observe actions constitutes a complete mission.



(a)

(b)

(c)

Fig. 5 An example of pedestrian detection

3. Assessment Methodology

All parts of the capstone assessment presented here took place in October 2014 at the Combined Arms Collective Training Facility (CACTF) at Fort Indiantown Gap, PA. The CACTF is composed of 13 buildings and 5 streets, representative of a neighborhood of a few blocks. An overhead view of the CACTF is presented in

Fig. 6, with 3 buildings used as test sites distinguished by labels. The church and the bar were the sites of most runs evaluating semantic navigation and perception, and of all runs testing door and pedestrian detection. The gas station was also used as a test site for semantic navigation and perception. The space between the gas pumps and the building behind it is used as a storage area for metal lockers, concertina wire, and metal barrels, providing a cluttered environment within which perception was difficult. The church, by contrast is a simple building without clutter, except for trash cans in one front corner. It stands apart from other buildings, is made of cinderblocks, and has tall windows with gray wooden shutters, and gray doors. The bar is not as cluttered as the gas station, but has a more complicated façade, with doors and windows set back from the street by several feet. It is also surrounded by other buildings, requiring the semantic navigation system to use landmarks to help select the correct building.

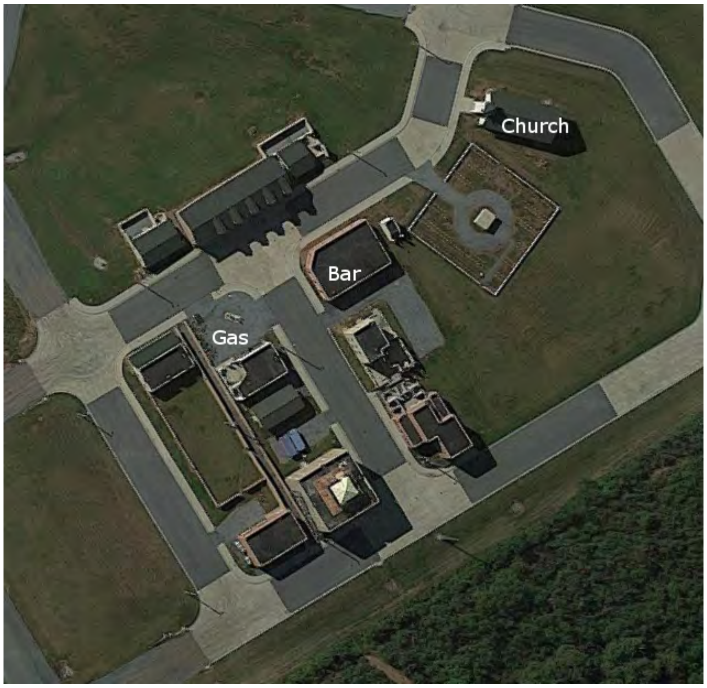


Fig. 6 Overhead view of the CACTF

3.1 Experimental Design for Semantic Navigation and Perception

Evaluating semantic navigation and perception required determining how well the system could navigate when given directions in terms of relative locations and semantically described goals and landmarks. The assessment described here evaluated system performance in executing 30 TBS commands. The 30 commands given to the system are contained in Table 1. These commands were divided into 12 different vignettes, which varied the 1) building, 2) orientation of the system,

3) number and type of semantic objects, 4) navigation constraints, and 5) the presence of clutter. Vignettes and associated TBS runs were developed to leverage the physical constraints of the CACTF, while providing varying levels of difficulty for semantic perception and navigation within the context of the screening mission. The design involved 2 replications of each of the 30 commands, but on different systems, operating in parallel at different test sites, for a total of 60 planned runs. The design was structured, but in statistical terms was necessarily unbalanced in order to provide more data emphasis to specific scenarios that had proven challenging in previous assessments. With limited time on the CACTF, only 54 of the planned 60 runs were conducted, although each of the 30 TBS commands was tested at least once.

Table 1 TBS commands for all runs[a]

Vignette	Runs	Tactical Behavior Specification by Run (Run in Parentheses)
1	1–3	Navigate to the traffic barrel near the building (1)/car (2)/fire hydrant (3).
2	4–5	Navigate to the traffic barrel near the building (4)/car (5)/fire hydrant (6).
3	7–9	Stay left (7)/right (8,9) of the building; navigate to a traffic barrel that is left of (7,9)/behind (8) the building.
4	10–11	Stay left (10)/right (11) of the building; navigate to a traffic barrel that is left of (10)/behind (11) the building.
5	12–13	Stay left (12)/right (13) of the building; navigate to a traffic barrel that is left of (12)/behind (13) the building.
6	14–15	Stay right of the car; navigate to a traffic barrel that is behind (14)/left of (15) the car (14)/building (15).
7	16–17	Stay left (16)/right (17) of the traffic barrel; navigate to a fire hydrant that is behind the building.
7	18	Stay right of the fire hydrant; navigate to the right of the building that is near the fire hydrant.
8	19	Stay right of the fire hydrant; navigate to a car that is behind the building.
8	20	Stay left of the fire hydrant; navigate behind the building that is behind the fire hydrant.
9	21	Stay right of the car; navigate to the left of the building that is near the fire hydrant.
9	22	Stay right of the traffic barrel; navigate behind the building that is near the traffic barrel.
9	23	Stay left of the traffic barrel; navigate to the fire hydrant that is left of the building (and left of your current position was added for one replication).
10	24	Navigate to the left of the gas pump.
10	25–26	Stay left of the gas pump; navigate to the building (25)/traffic barrel (26) that is left of the gas pump.
11	27	Navigate to the building that is right of the car.

Table 1 TBS commands for all runs[a] (continued)

Vignette	Runs	Tactical Behavior Specification by Run (Run in Parentheses)
11	28	Navigate to the building that is left of and behind the car.
12	29	Stay right of the car; navigate to the building that is behind the car.
12	30	Stay right of the car; navigate covertly to the building that is behind the car.

[a] Within most vignettes, the basic structure of the command was the same, with goal and reference objects changing. When this is true, the runs during which the particular goal or reference object was used is in parentheses behind the object. For example, "building (1)/car (2)/fire hydrant (3)" means the relevant object was building for run 1, car for run 2, and fire hydrant for run 3.

3.2 Evaluation of Semantic Navigation

Semantic navigation was evaluated by human observers, who graded each run on a scale of 0% to 100%, with gradations of 20% (i.e., 0, 20, 40, 60, 80, 100). The scored was based on how complete the observer perceived the run to be, without taking into account the technical specifications of the robot. Thus, whether the robot considers itself close enough to an object to be finished, or whether the robot is acting based on a perception of the world different from that of the human observer, is not considered. These grades were primarily used to develop high-level quantitative summaries. The scores were also treated qualitatively, and reported in terms of frequency distribution to clarify findings. Decoupling the human evaluation from technical considerations may occasionally penalize or reward robotic behavior that could easily be different with different parameter settings. However, since the robot is intended to eventually work with human teammates, the final evaluation of its performance is based on the opinions of humans standing in for those teammates.

As the scoring was performed by 2 separate observers, evaluating separate (but identically configured) robots, there was a concern as to how similarly scoring criteria would be applied. To alleviate this concern, the test director moved back and forth between the 2 observers to verify that their grading was consistent enough to be considered jointly, rather than analyzing the scores by observer. The 2 observers repeated the same trials to the extent time allowed, and here we consider those repeated trials as replications. Thus we consider it reasonable to subsequently analyze the scoring without reference as to which observer did the scoring.

3.3 Evaluation of Perception

Semantic perception has been previously evaluated in Lennon et al.[4] There, the labeling of each pixel within an image, by an algorithm, was compared with similar labeling done by a human, and the algorithm was graded based on the number of pixels correctly labeled in each image. Pixel-based evaluation is limited, however, in that it does not take account of how errors in the labeling affect the intelligence using the labeling. For example, consider Fig. 7, representing 2 images (white rectangles) in which some pixels have been correctly labeled as car pixels (dark blue) and others pixels are incorrectly labeled as car (light blue). In the left image, the intelligence architecture using this perceptual system would still understand there to be one car, although larger than the true one, while from the right image, it would understand there to be 2 separate cars. These distinct errors would be graded as the same as long as the same number of pixels are incorrectly labeled, but would have different effects on how the intelligence architecture relates to the world.

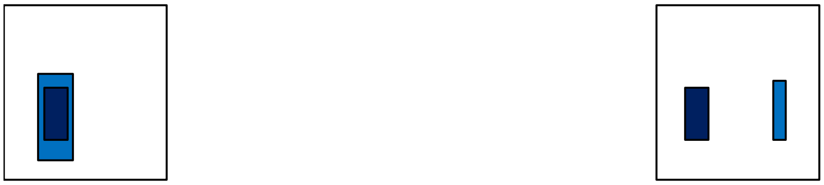


Fig. 7 An example illustrating a limitation of pixel-based evaluation

An evaluation of perception as part of an integrated system should take into account how the intelligence architecture is using the perceptual information, which is an indication on overall system performance. The intelligence architecture that plans semantic navigation was relying on perception to populate its model of the world, based on which model it developed a plan for accomplishing its mission. Consequently, we intend here to evaluate semantic perception based on how closely the world model it populates represented the actual world to which the robot was exposed. This method will be used to evaluate the labeling of small objects. The labeling of terrain features was not used as part of semantic navigation and was assessed in Lennon et al.,[4] while the labeling of smaller objects was not, since such capabilities had not been implemented at that time. For this assessment, we evaluated the capability of the Carnegie Mellon University (CMU) and University of Pennsylvania (UPenn) systems to identify traffic barrels and fire hydrants and of the UPenn system to identify cars and gas pumps. This evaluation was based on the number of false positive and false negative object detections, and on the number of times false positive detections occurred.

3.4 Design and Evaluation of Door and Pedestrian Detection

The assessments of door detection, and of the detection and tracking of pedestrians, were divided between the church and bar. Nine platform locations at the church, and 8 at the bar, were arranged in a lattice with rows starting at the middle and ends of the buildings, and additional columns spaced evenly at 10-m intervals. The incomplete lattice at the bar was due to the close proximity of a neighboring building. The layouts for these designs are shown in Fig. 8. For the façade feature detection, 3 initial orientations were used at the church: platform pointed parallel to the building, away from the building, or in between the 2 buildings. At the bar, 3 orientations were used for some positions and 2 for others, with the "in between" orientation being omitted in those cases. This was done to compare the numbers of doors detected at the church and bar. The church had 2 doors to detect and the bar had 3; therefore, different numbers of orientations permitted each building to provide 54 possible door detections. In Fig. 8, the black dots at the bar represent positions at which 3 orientations were used, and 2 orientations were used at all other bar positions. The red dots represent the positions at which pedestrian detection was evaluated. Columns are spaced at intervals of 10 m, and the angles represent the initial orientation of the robot for door detection. The number of doors detected at each position was recorded for evaluation.

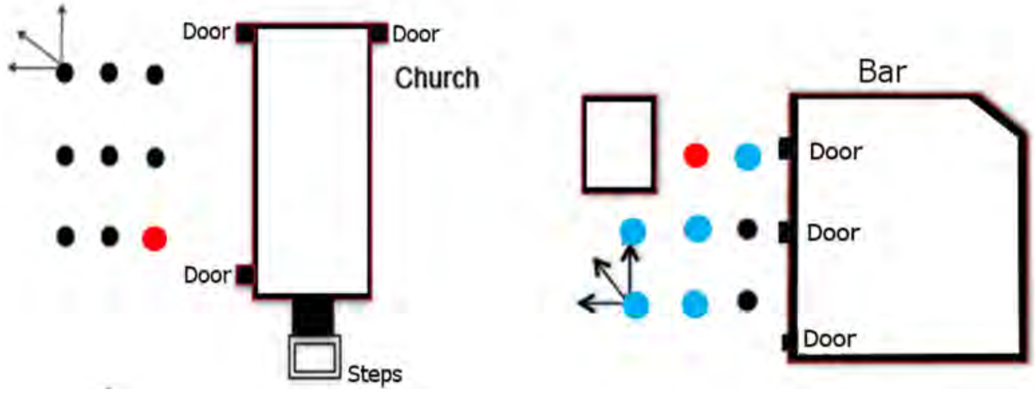


Fig. 8 Layout for door detection

The pedestrian search was conducted with only an initial orientation toward the building. Pedestrians emerged from building doors, or from a standing position in close proximity to building walls, and then advanced in a variety of coordinated movements, including parallel in the direction of the platform, parallel perpendicular to the direction of the platform, and crossing or fanning out movements. For half the runs, traffic barrels were placed so that they would partially obscure one or more pedestrians at some point during their route. Two or 3 pedestrians appear in each run, and the robot was evaluated on how many correct detections it made.

4. Results

4.1 Results for Semantic Navigation

Within this section, the completion scores are broken down in tabular form over a variety of test conditions. A brief interpretation of each table provides insights as to the conditions under which the technology performed well, and the conditions that proved challenging. The completion score is the subjective assessment of the degree to which the platform accomplished the mission. In Table 2, the subjective scores are treated quantitatively for a concise representation of performance over the 12 vignettes. Subsequent tables are presented as frequency distributions for the subjective scores, to provide greater detail for individual run performance, both in aggregate and against specific test conditions. Table 3 presents the score for each run and states the evaluator’s reason for assigning a score less than 100%.The average score over each vignette is presented in Table 2, which also includes details about the vignette.

Table 2 Vignette descriptions

Vignette No. and Description		Location	No. Runs	Avg. Score (%)	Runs < 100[a] (%)	TB/FH/Car[b]
1	In front with no clutter	Church	6	90	80, 60	3/1/1
2	In front with clutter	Church	6	100	…	3/1/1
3	Single building and no clutter	Church	6	90	80 (3)	3[c]/1/0
4	Single building and clutter	Church	4	90	80 (2)	3[c]/1/0
5	Single building from the side	Church (side)	4	65	60 (2), 40	3[c]/0/0
6	Around a car and building	Church	4	65	60 (2), 40	3/2/2
7	Choosing between buildings	Bar	3	73	60 (2)	2/2/0
8	Choosing between buildings	Bar	3	73	80, 40	0/1/2
9	Choosing between buildings	Bar (side)	6	90	80 (3)	1/2/1
10	Around a gas pump	Gas station	6	93	60	2/0/0
11	Around a car	Gas station	4	80	60 (2)	0/0/1
12	Around a car 50 meters	Bar to church	2	90	80	0/0/1

[a] For the column Runs < 100, score (x) indicates x number of runs with that score.

[b] The TB/FH/Car column indicates the number of traffic barrels, fire hydrants, and cars in the vignette.

[c] Either 3 or 4 appeared, depending on availability.

Table 3 Scores by run

Vignette	Run	Score 1/2	Reason for Scores Below 100% for Replications 1/2
1	1	80/100	False barrel detection led it up the steps where it was stopped for safety/NA
1	2	60/100	Robot stopped on the side of the car opposite the barrel/NA
1	3	100 100	NA/NA
2	4	100/100	NA/NA
2	5	100/100	NA/NA
2	6	100/100	NA/NA
3	7	100/100	NA/NA
3	8	80/80	Stopped early on route to barrel/Stopped early on route to barrel
3	9	80/100	Found barrel but did not stop/Stopped early on route to barrel
4	10	80/100	Stopped too far from barrel/NA
4	11	80/100	Stopped too far from barrel/NA
5	12	60/100	Did not stop at the barrel/NA
5	13	40/60	Went up the steps and stopped for safety/Battery died during mission
6	14	60/100	Went behind the car, then turned around and came back/NA
6	15	40/60	Classified car as a barrel/Classified car as a barrel
7	16	60	Went past the fire hydrant and did not see it
7	17	60	Went past the fire hydrant and did not see it
7	18	100	NA
8	19	40/80	Robot collided with building/Went past the car
8	20	100	NA
9	21	100/100	NA/NA
9	22	80/80	Stopped before reaching the back/Stopped before reaching back
9	23	80/100	Did not see the fire hydrant/NA
10	24	100/60	NA/Stopped in front of the gas pump
10	25	100/100	NA/NA
10	26	100/100	NA/NA
11	27	60/100	Stopped in front of vehicle, did not go around./NA
11	28	100/60	NA/Stopped early after seeing a hill as an obstacle
12	29	100	NA
12	30	80	Correct navigation, but nothing covert about it

Notes: Most runs were replicated, and the score column shows the scores for the first/second runs when replication occurred. For replications in which the score was below 100%, the reason for the score is recorded. Otherwise, the record is NA (not applicable).

In vignettes 1 and 2, the mission was a direct go-to route, with a semantically identified goal within line of sight of the start point. The goal for these vignettes was always a traffic barrel, but multiple traffic barrels were in the scene, along with a vehicle, church building, fire hydrant, and cemetery wall. These other standard objects in the scene forced the system to choose from among several possible objects and provided goal references. For example, in vignette 1, run 3, the TBS was “navigate to the traffic barrel near the fire hydrant”. The clutter introduced in vignette 2 required the robot to avoid obstacles that were below camera height, so as not to interfere with semantic classification. The average score results suggest a high level of performance in these basic vignettes, with 10 of 12 runs achieving the goal 100% consistent with expectations. In 1 of 2 unsuccessful runs, a false detection for the traffic barrel goal near the door of the church caused the system to approach the door, and the Husky platform was stopped while attempting to climb the steps. In the second, the platform stopped on the sidewalk, near, but on the other side of a vehicle from the goal.

Vignettes 3 and 4 were more challenging, in that the goal was not within line of sight for the semantic classifier. These 2 vignettes moved the mission closer to the screen-the-back ideal, where the existence of the identified goal is initially supported only by the trusted command of the operator and must be discovered by the system as it semantically navigates to the described location. Intermediate directions, commanding the system to move around a specific side of the building on the way to the goal, added complexity. Further, these runs called for a greater distance to be traveled, with more opportunities for error as the system interacted with the standard objects and the building. The average scores, 90% for each vignette, again show a high level of performance. Of the 5 runs receiving an 80% score, 4 stopped just short of the goal and 1 turned away from the goal toward the end of its route and traveled off course.

Vignettes 5 and 6 changed the initial perspective of the robot to challenge its sense of what the front of the building was. Vignette 5 started the platform on the left-side wall of the church, facing that wall. In this position, what was formerly the left side of the church would now be interpreted as the front. From this position, when asked to go to the left, the robot found the traffic barrel goal in one run (100%) but completely missed it in another (40%). When asked to go around the church to the right, the robot had difficulties in navigating around the porch in front of the church and in getting around the church (60% for both). Some of these difficulties may have been because the XR LADAR sensed buildings in the distance, creating ambiguity as to which building the robot was to go behind. Vignette 6 offset the platform, still in the front of the church, but at such an angle that both the front and right side of the church could be seen by the XR LADAR, thus introducing ambiguity in the building orientation. In vignette 6, cars served as intermediate direction references (e.g., stay left of the car). There was no line of sight to the goal for any of these runs. The traffic barrel behind the vehicle was found each time, but in one run, the platform doubled back to the start rather than remaining at the goal (100% and 60%). The runs sending the robot to the left of the church both failed because the robot mislabeled a vehicle as the traffic barrel goal (40% and 60%).

Vignettes 7 through 9 leveraged the more complex building structures around the bar. At the bar, 2 buildings were always in the initial view, the bar and one other. To successfully complete 3 of the 8 planned runs, the robot needed to interact with at least 3 buildings. In addition, intermediate directions for navigation and goal references were expanded to include any of the standard objects. For example, in run 16, the TBS was “stay left of the traffic barrel; navigate to a fire hydrant that is behind the building”. Only one run could be considered as having line of sight to the goal from the starting position. Two of the 3 runs in vignette 7 received scores of 60%. In one case, the robot did not see the fire hydrant. In the other, it is likely that the building predictor sized the building as bigger than its actual size, thus limiting the ground for an acceptable path to the goal. In vignette 8, 2 runs had difficulty clearing the building to the right. The likely cause was inadequate building prediction resulting from an irregular corner of the building (a corner porch at 45°). It appeared that when the robot encountered the first corner, it interpreted the “end” of the building as having been reached and then turned into the angled corner of the building. Of the 3 runs in vignette 9 that scored 80%, the fire hydrant was not detected in one, and in the other 2, the robot did not get behind the building to the goal. It is likely that the building predictor placed the back wall in the wrong location, so that navigation thought the robot was “behind” the building when it was actually still out in the open.

Vignettes 10 and 11 introduced gas pumps as an additional semantic object, together with the bar and other background buildings. Five unique runs were planned, and each of the 5 was completed in at least 1 replication. Most runs (7 of 10) were scored at 100%. Vignette 10 focused on the gas pumps as intermediate direction and goal references. The goal area was visible from the starting position in the first 2 runs of vignette 10. The 60% run occurred when the robot stopped in front of the gas pumps instead of to the left side. In vignette 11, the gas pumps were in the scene, but the focus was reaching buildings relative to a vehicle. In one 60% run, the robot stopped at the vehicle goal reference and did not proceed to the goal. In the other 60% run, the robot was planning to the correct building but did not try to overcome a hill along the way.

Vignette 12 started the robot in front of the bar, and directed that it travel covertly or quickly around a car to the church in the distance. In both cases, the robot traveled around the car to the church as instructed. The 80% run asked for covert movement, which in this instantiation involved simply moving the robot to the side of the street, on the periphery of the scene. No covert action was evident.

Tables 4–6 provide a further breakdown of the results, and complement the above discussion. Here, the data was parsed according to specific test conditions. Each run was characterized in terms of the specific building, line of sight, clutter, goal reference, and goal. In Table 4, the frequency in obtaining each score is broken out by building or site within the CACTF. This is an aggregation of the previous description of the vignettes. Vignettes 1–6 occur at the church, vignettes 7–9 occur at the bar, vignette 10 occurs at the gas pump, and vignettes 11–12 are considered multiple buildings. We see that 30 of 54 of the runs resulted in a perfect score and only 13 of 54 scored less than 80%. Those runs with poorer performances occurred in approximately the same percentage at the bar and church locations.

Table 4 Score distribution by site

Score	Bar	Church	Gas pump	Multiple	All
40	1	2	0	0	3
60	2	5	1	2	10
80	4	6	0	1	11
100	5	17	5	3	30
All	12	30	6	6	54

Table 5 highlights the impact of clutter and line of sight. Clutter comparison was performed on pairs of runs, differing only in the presence of clutter, over the first 4 vignettes at the church. There was no indication that clutter hindered the robot's ability to complete the mission. A comparison based on whether the goal was within the initial line of sight of the robot is also shown. Both conditions were included at the church, bar, and gas pump. The table suggests that when the goal is within view at the start, the likelihood of getting there successfully is much greater.

Table 5 Score distribution by clutter and line of sight

Score	Clutter			Line of Sight		
	No	Yes	All	Yes	No	All
40	0	0	0	0	3	3
60	1	0	1	2	8	10
80	3	2	5	1	10	11
100	6	8	14	15	15	30
All	10	10	20	18	36	54

Table 6 shows the average score and number of runs for each goal reference by goal combination. Goal reference describes the goal relative to another object's position. The goal is the object to be reached. In aggregate, we see that the average score was 85% over the 54 runs. The marginal average percentages do not suggest any serious problem with goal references or goals. It is was potentially interesting to look at the average percentages in the margin associated with a specific object, for example fire hydrants as a goal yielded a 75% average score, and as a goal reference yielded a 100% average score; however, there were no definitive patterns seen.

Table 6 Average percentage score and count by goal reference and goal

Goal Reference	Goal										All	
	Building		Fire Hydrant		Gas Pump		Traffic Barrel		Vehicle			
	%	No.	%	No.	%	No.	%	No.	%	No.	%	No.
Building	…		75	4	…	…	82	20	60	2	79	26
Fire Hydrant	100	4	…	…	…	…	100	4	…	…	100	8
Gas Pump	100	2	…	…	…	…	100	2	…	…	100	4
NA	…	…	…	…	80	2	…	…	…	…	80	2
Traffic Barrel	80	2	…	…	…	…	…	…	…	…	80	2
Vehicle	83	6	…	…	…	…	87	6	…	…	85	12
All	90	14	75	4	80	2	86	32	60	2	85	54

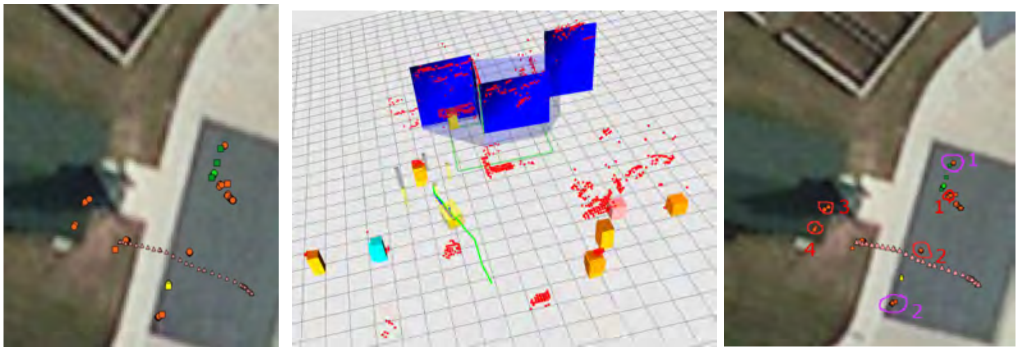
Note: NA = not applicable

4.2 Results for Semantic Perception

Evaluating perception based on the robot's world model presents several challenges. First, due to difficulties in localization, the robot will not know precisely where it is at all times. Since the perceptual system records object positions relative to the robot's position, this means that errors in localization will lead to objects perceived to be at incorrect locations. As we currently have no accurate ground truth for robot position (GPS can be off by 10 m), we cannot always distinguish errors in localization from errors in perception, at least in terms of object location. Thus, an object recorded as several meters from the closest ground truth object of its class could be a false positive, or it could be an error in localization. Likewise, if a ground truth object has no perceived objects near it, that could be a false negative, or it could be a localization error, with the object having been perceived but recorded at a different location. As a consequence, evaluations of perception based only on the distances of perceived objects from ground truth objects will not be reliable until we have a better method of establishing ground truth for the position of the robot and the objects.

However, we do have the ability to play back the robot's construction of its world model as it runs through the mission. Combined with plotting the ground truth and perceived object locations on a map, this playback can give us some understanding of what the robot perceived. Consider Fig. 9, which shows, from left to right, 1) a plot of perceived traffic barrel positions (as orange discs), 2) an image from the world model, and 3) an evaluated version of the leftmost plot. The legend describing all symbols in the figure is in Table 7. Of the barrels plotted on the leftmost map in Fig. 9, which should be considered correct, which should be attributed to localization error, and which should be counted as perception errors?

Based on viewing the playback from the world model, one can discern that the robot did indeed perceive 2 different traffic barrels in the line between it and the car, and the one the evaluator determined to be incorrect is circled in red and labeled with the red number 1. Similarly, one can ascertain from the relative positions of objects simultaneously perceived that the other groups of barrels circled in red were false positives for the CMU semantic classifier. There were 4 false positive objects detected during the run, with no false negatives for the CMU classifier. For the UPenn classifier, there was one false positive traffic barrel and one false negative traffic barrel, indicated by the purple circles numbered 1 and 2, respectively.



(a) (b) (c)

Fig. 9 Map (a and c) and world model image (b) from run 4

Table 7 Legend for plots of objects shown in Fig. 9.

Object	Marking (Fig. 9a and c)	Object	Marking (Fig. 9b)
Traffic barrel (robot)	Orange disc	Traffic barrel (human)	Orange square
Car (robot)	Green disc	Car (human)	Green square
Fire hydrant (robot)	Yellow disc	Fire hydrant (human)	Yellow square
Gas pump (robot)	Light blue circle	Robot position	Pink triangle
CMU classifier error	Marked in red	…	…
UPenn classifier error	Marked in purple	…	…

Notes: Objects could be identified by either a human observer during setup (human) or by the robot during the course of the run (robot). In general, human-identified objects are squares, and robot-identified objects are discs.

Simply denoting the number of falsely perceived objects gives an incomplete picture of the system's perceptual abilities, however. The classifiers are running during the entire mission, and there may be a difference between an object (e.g., a box) being classified as a traffic barrel once in passing or for the duration of the mission. To examine these different types of errors, we enumerated how many times the misclassification of each different object type occurred. For example, the left side of Fig. 10 shows a plot of traffic barrel observations during run 22. The CMU classifier detected a traffic barrel 303 times, and 59 of these were the false detection of the 1 traffic barrel circled in red. The classifier only detected one false object, but it was quite "certain" that the object was there. Meanwhile, the right side of Fig. 10 shows a plot of traffic barrel observations during run 24. The CMU classifier detected a traffic barrel 991 times, and 231 of these were the 8 false detections of the traffic barrels circled in red. There were more incorrectly classified objects in run 24, but some were incorrectly classified for only a brief period of time. In fact, 3 of the false positive traffic barrels were identified as barrels fewer than 10 times. There are other errors indicated in the images, and these are discussed, along with errors for the other runs, in the Appendix.

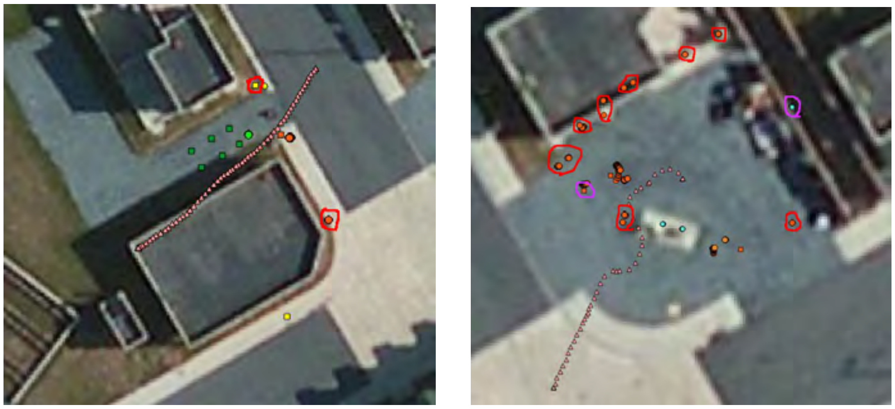


Fig. 10 Plots of the perception results from run 22

We evaluated perception within 4 vignettes. We chose vignettes 2, 3, 9, and 10, in each of which there were 3 runs, with no objects being moved between runs. Two of these vignettes were at the church, one at the bar and one at the gas station. We faced some challenges in creating the maps used for the evaluations. First, in the course of the testing, the robot began each run with an unknown orientation, and without a precisely mapped out initial position. Thus, the robot's world model needed to be translated and rotated to correspond with maps of the world, and the exact parameters of this transformation are unknown. An evaluator needed to choose a rotation and translation based on which to evaluate the data. It should be noted that there is no reason to believe that semantic perception would be biased in determining the angle or distance of objects from the system, so we are not concerned about these properties being altered by our method of aligning the data. Second, under the time restrictions of testing, the precise mapping of every perceivable object would have led to a reduction in the number of runs we could perform, and was not deemed to be worth the cost. As a compromise, the positions of objects were sketched on a map by the evaluators, leading to some lack of precision in the markings of the object locations. We have concluded that the precision is enough, however, for the method of evaluation we are using, in which objects are determined to be correctly or incorrectly detected based on their relative positions, with playback of the world model used for verification.

The CMU classifier and UPenn object detector were employed to work together on the robot as complements, with the CMU classifier labeling everything within the environment as one of the classes described in Section 2.2, and the UPenn object detector looking only for traffic barrels, fire hydrants, cars, and gas pumps.

Table 8 contains the ratio of false positives and false negatives to the true numbers of objects visible to the robot (for traffic barrels and fire hydrants). For fire hydrants, the CMU classifier tended to miss them at a higher rate than the UPenn detector. The CMU and UPenn systems missed one-half and one-quarter (respectively) of the observable fire hydrants. Meanwhile, the UPenn detector missed a substantial number of traffic barrels, while the CMU classifier substantially overcounted. As mentioned previously, however, with regard to false positives, we should consider not just how many false objects were detected, but how many of the object detections were false.

Table 8 Ratio of false positives and negatives to true number of objects visible for traffic barrels and fire hydrants

Classifier System	Traffic Barrels		Fire Hydrant	
	False Positive	False Negative	False Positive	False Negative
CMU	39/50 (0.78)	0	0	(8/15) 0.533
UPenn	3/50 (0.06)	(11/50) 0.22	0	(4/15) 0.267

Considering Fig. 11, we see that, although the detection at least 2 of false traffic barrel objects does imply that at least 10% of detections were false, there is no linear relationship evident between the number of detected objects and the ratio of false detections. The CMU classifier may see a large number of false positive objects, detecting them only sporadically, or a few false positive objects that it repeatedly detects as traffic barrels. As an example, we reconsider the right image of Fig. 10, which shows 8 falsely detected objects around the gas station. We counted how many times each object was falsely detected for this run. For the CMU classifier, the number of times the 8 traffic barrels were falsely detected were (from high to low): 95, 38, 33, 27, 25, 9, 2, and 2. Many of these detections would have occurred while the robot was sitting still, and it is not surprising that an object falsely detected once from a stationary position would have been repeatedly falsely detected from that position. Those objects detected only a few times are like noise, objects that briefly appear, and then disappear. The UPenn detector appears less susceptible to such noise, although more prone to miss the traffic barrels. The current strategy of using 2 object detectors for some objects, seems to have worked well, give the results of the semantic navigation (see Section 4.1), and the CMU and UPenn detector seem to complement each other. Although running multiple perception algorithms is computationally expensive, it appears necessary given the current state of semantic perception, and the cost is likely worth it when compared with the cost of basing planning on faulty perceptual information. Still, even with 2 perception systems running, at least 6 runs were unsuccessful due primarily to perception difficulties (see Table 3), and other failures may have been in part the result of perception issues. Moreover, in the case of traffic barrels and fire hydrants, these perception systems were working together to identify objects that were not occluded and clearly distinguished from the background (clearly to a human). This suggests that, even with continuing progress in perception, we should assume that semantic perception will still require substantial computational resources to perform object identification online.

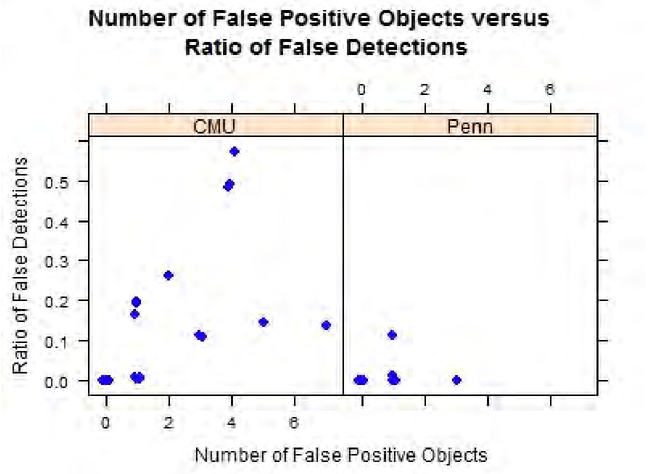


Fig. 11 Number of false traffic barrel objects detected vs. the ratio of false detections/total detections of traffic barrels

4.3 Results for Door and Pedestrian Detection

For the search activity at the church, there were 54 doors to be discovered: 3 platform orientations by 9 lattice positions by 2 doors. For the bar, there were also 54 doors to be discovered: 18 orientation and position combinations by 3 doors (see Fig. 8). High-level results for the search and observe portions of the preliminary experiments are reported in Table 9.

Table 9 Evaluation of door and pedestrian detection by location

Location	Doors Found[a]	Door Misses L/M/R[b]	Door Misses C/M/F[c]	Pedestrians[a]
Side of the church	48 of 54 (89%)	5/NA/1	2/3/1	21 of 24 (88%)
Back of the bar	45 of 54 (83%)	0/4/5	5/4/0	15 of 27 (56%)

[a] The Doors Found and Pedestrians columns show the number of doors (pedestrians) detected out of the number presented in the scene, and the percentage.

[b] The Door Misses L/M/R column shows the number of doors missed on the left/middle/right of the building.

[c] The Door Misses C/M/F column shows the number of doors missed in the closest, middle, and farthest search distances.

Among the 9 doors missed at the bar, the middle door was missed 4 times and the right-most door was missed 5 times. The misses were split evenly between the parallel and in-between initial orientations, with one miss resulting from the robot initially pointing away. Thus, no real difference can be claimed due to initial orientation; however, the clockwise turn of the robot swept left to right over the building façade, so more cycles were spent on doors to the left. Not surprisingly, among the 3, the left-most door was almost always identified. All runs resulting in missed doors at the bar were at the closest 2 rows to the bar façade. The 3-D orientation of the platform turning on the stone surface, coupled with the close distance to the building, may have prevented the sensor from receiving information above the doorframe height, thus providing an incomplete match with the door template. Fewer doors were missed at the church. Of those missed, 5 of 6 came from the 2 rows of starting positions closest to the façade. From a recall standpoint, the previous discussion suggests doors that are on the target building are found reliably. From a precision standpoint, the system often classified structures in the façade of nearby buildings, for example windows, as doors, although these instances were not recorded reliably enough for analysis. Detection of actual doors on buildings other than the target building was not a concern, as the intent would be that the system search for doors in the target building identified as part of semantic navigation, rather than accept any nearby door or window.

Pedestrian detection was highly successful at the church, missing only 3 pedestrians out of the 24 presented. Each of the 3 not detected resulted from runs where pedestrians were executing a fanning out movement toward the platform. It is likely that the identifiers for the pedestrians in the scene were not seen over enough cycles to be recognized before the pedestrians had closed on the platform. Some attempt to mitigate this result had been established in the protocol. Pedestrians were to remain motionless for a few seconds before starting their route so that the system could acquire them. At the bar, 12 pedestrians were missed out of 27 presented. Among the 12, 8 were missed in movements toward the platform (7 fanning and 1 crossing movement). The other 4 involved movements along the façade wall away from the platform. The issue with both movement types could be the number of cycles that it takes to recognize the object as a person. Moving out of the platform's 10- to 15-m detection range could have led to the 4 misses of pedestrians moving down the façade. In addition to these observations, clutter that partially obscured the detected pedestrians often broke the track. The pedestrians reappeared on the other side of the clutter, but the pedestrian was moving, and the system was unable to reestablish the track.

5. Conclusions

In this study, we were able to assess important components of an unmanned ground system autonomously screening the back of a building for pedestrians. We considered the semantic recognition of objects, including buildings and structure along a proposed route, and navigation with respect to those objects. We also evaluated the ability to recognize important structures on the façade of a building where people on foot were likely to emerge, in addition to identifying and tracking pedestrians over a short distance. Generally, the architecture governing semantic perception and navigation seems to be working well. There is need for the expansion of semantically classified objects and greater reliability in classifying the included objects. False detections introduced into the world model hurt performance in some runs. However, a comparison of the current results (Appendix) with those obtained during a previous integrated research assessment[10] suggests that perception is functioning more reliably as part of the integrated system. The introduction of the UPenn object detector as a complement to the CMU semantic classifier likely contributed to the improved performance of the system. We would expect improvements in each of these components, and in the fusing of their results, to further improve performance. In movements relative to a building, building prediction as to size and orientation created situations where the robot navigation, though technically successful, would not be consistent with user expectations. Again, however, comparison with the results in Lennon et al.[10] suggests that building prediction is also functioning more reliably as a system component. The increase in sensor range introduces more ambiguity regarding which building or object is being referred to. In a parallel effort, HRI advances over voice, gesture, and text are being developed to support negotiation with the robot to reconcile ambiguities that are always going to be present to some degree.

In door detection, the system performed reasonably well. However, improvements could be made with respect to the distances at which the detections could reasonably be expected to occur. Also, a more complete sweep of the façade to detect doors that were not sensed often enough to be detected would improve the rate of success. Pedestrian detection performed well when the pedestrians stayed in position long enough for the initial detection to take place. Improvements to the speed of this initial detection would likely improve the overall performance of pedestrian detection and tracking.

The capstone represents the conclusion of the first 5 years of RCTA research, but also serves as a baseline for the next 5 years. We expect these years to include improvements of specific components and generalization of the intelligence architecture to incorporate a wider range of capabilities. The basic capabilities and intelligence architecture of the system seemed sound, at least within the narrow range of scenario in which it was tested. We expect that the next 5 years will include generalization of the capabilities of the system to a variety of scenarios and possibly the ability to learn from repeated training within a scenario, as well as closer interaction with human teammates.

6. References

1. Robotics Collaborative Technology Alliance. Robotics Collaborative Technology Alliance (RCTA) FY 2012 annual program plan. Aberdeen Proving Ground (MD): Army Research Laboratory (US); 2012 Mar. [accessed 2015 12 March]. http://www.arl.army.mil/www/pages/392/RCTA_FY12_APP.pdf.

2. Hill S, Barber D, Evans A. Achieving the vision of effective soldier-robot teaming: recent work in multimodal communication. HRI'15 Extended Abstracts – Proceedings of the Tenth Annual ACM/IEEE International Conference on Human-Robot Interaction Extended Abstracts; HRI'2015; 2015 Mar 2–5; Portland, OR. New York (NY): The Association for Computing Machinery; c2015. p. 177–178.

3. Munoz D. Inference machines: parsing scenes via iterated predictions [Ph.D. thesis]. [Pittsburgh (PA)]: The Robotics Institute, Carnegie Mellon University; 2013.

4. Lennon C, Bodt B, Childers M, Camden R, Suppe A, Navarro-Serment L, Florea N. Performance evaluation of a semantic perception classifier. Aberdeen Proving Ground (MD): Army Research Laboratory (US); 2013 Sep. Report No.: ARL-TR-6653. Also available at: http://www.arl.army.mil/www/default.cfm?technical_report=6868.

5. Oh J, Suppe A, Duvallet F, Boularias A, Vinokurov J, Navarro-Serment L, Romero O, Dean R, Lebiere C, Hebert M, Stentz A. Toward mobile robots reasoning like humans. Proceedings of the 29th AAAI Conference on Artificial Intelligence; 2015 Jan 25–30; Austin, TX. Palo Alto (CA): Association for the Advancement of Artificial Intelligence; c2015. p. 1372–1379.

6. Zhu M, Atanasov N, Pappas GJ, Daniilidis K. Active deformable part models inference. In: Fleet D, Pajdla T, Schiele B, Tuytelaars T, editors. Computer Vision – ECCV 2014, Proceedings Part VII. ECCV 2014 – 13th European Conference on Computer Vision; 2014 Sep 6–12; Zurich, Switzerland. Cham (Switzerland): Springer International Publishing; c2014. p. 281–296.

7. Dean R. Common world model for unmanned systems. Proc. SPIE 8741. Unmanned Systems Technology XV. 2013 May 17. doi: 10.1117/12.2016606.

8. Yang Y, Ramanan D. Articulated pose estimation using flexible mixtures of parts. 2011 IEEE Conference on Computer Vision and Pattern Recognition (CVPR); 2011 June 20–25; Providence, RI. New York (NY): IEEE. c2011. p. 1385–1392.

9. Rusu RB, Cousins S. 3-D is here: point cloud library (PCL). 2011 IEEE International Conference on Robotics and Automation (ICRA); 2011 May 9–13; Shanghai, China. New York (NY): IEEE. c2011. p. 1–4.

10. Lennon C, Bodt B, Childers M, Dean B, Oh J, DiBerardino C. Assessment of navigation using a hybrid cognitive/metric world model. Aberdeen Proving Ground (MD): Army Research Laboratory (US); 2014. Report No.: ARL-TR-7175.

Appendix. Perception Evaluation of Capstone Experiment

This appendix contains the evaluations of perception for 24 runs of the capstone experiment. The legend describing the objects in all the plots is contained in Table A-1. The description of the runs, including the tactical behavior specification (TBS) commands, is provided in Section 3, Tables 1 and 2. In some cases, runs needed to be repeated during experimentation due to platform failure, inference with the scene, or other reasons. This is recorded by a letter after the run. For example, run 23b is a repeat of run 23.

Table A-1 Legend for plots of objects

Object	Marking	Object	Marking
Traffic barrel (robot)	Orange disc	Traffic barrel (human)	Orange square
Car (robot)	Green disc	Car (human)	Green square
Fire hydrant (robot)	Yellow disc	Fire hydrant (human)	Yellow square
Robot position	Pink triangle	Gas pump (robot)	Light blue circle
CMU classifier error	Marked in red	…	…
UPenn classifier error	Marked in purple	…	…

In Figures A-1 through A-12, false positives and false negatives are indicated by red circles for the Carnegie Mellon University (CMU) classifier and by purple circles for the University of Pennsylvania (UPenn) object detector. Each figure shows the evaluations for one run, with the replication on Husky 1 on the left, and that of Husky 3 on the right. Following each figure, Tables A-2 through A-13 show the following information (from left to right):

1. The number of the Husky.
2. The classification/object detection algorithm.
3. The number and types of ground truth (GT) objects in the scene that would have been visible to the robot, (e.g., w T, x F, y C, z G means there were w traffic barrels, x fire hydrants, y cars, and z gas pumps in the scene).
4. The number of false positive (FP) object detections, using the same notation as for the true number of objects.
5. The number of false negative (FN) object detections, using the same notation as for the true number of objects.
6. The remaining columns show the fraction of false detections over the total number of detections for each object type, for traffic barrels (TBs), fire hydrants (FHs), cars, and gas pumps. If an object was not detected, the cell is “NA”. The CMU classifier was only used to identify traffic barrels and fire hydrants (in addition to the terrain types listed in Section 2.2), so the car and gas pump column are not relevant for this algorithm. The UPenn object detector was used to identify all 4 types of objects.

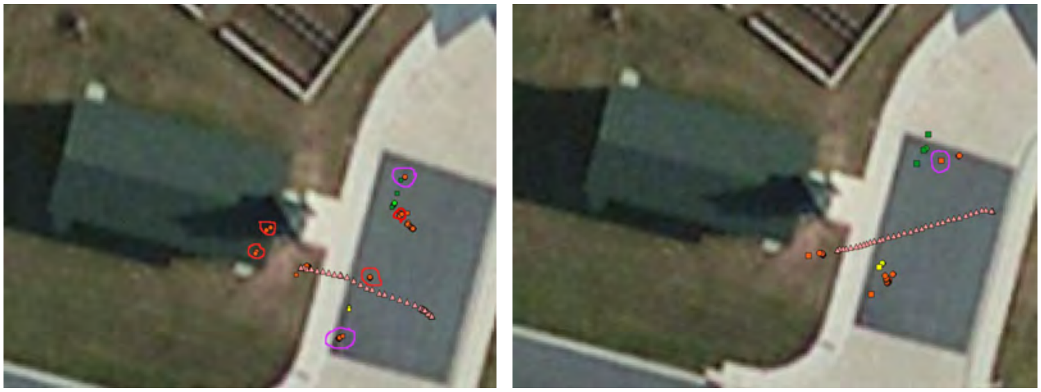


Fig. A-1 Run 4: left shows Husky 1, right shows Husky 3

Table A-2 Run 4 evaluation of perception

Husky	Alg.	GT Obj.	FP Obj.	FN Obj.	False/All TB Obs.	False/All FH Obs.	False/All Car Obs.	False/All Gas Obs.
1	CMU	3 T, 1 F	4 T	0	550/1,120	0/200	NA	NA
1	UPenn	3 T, 1 F, 1 C	1 T	1 T	7/63	0/14	0/47	NA
3	CMU	3 T, 1 F	0	0	0/420	0/45	NA	NA
3	UPenn	3 T, 1 F, 1 C	0	1 T	0/49	0/31	0/27	NA

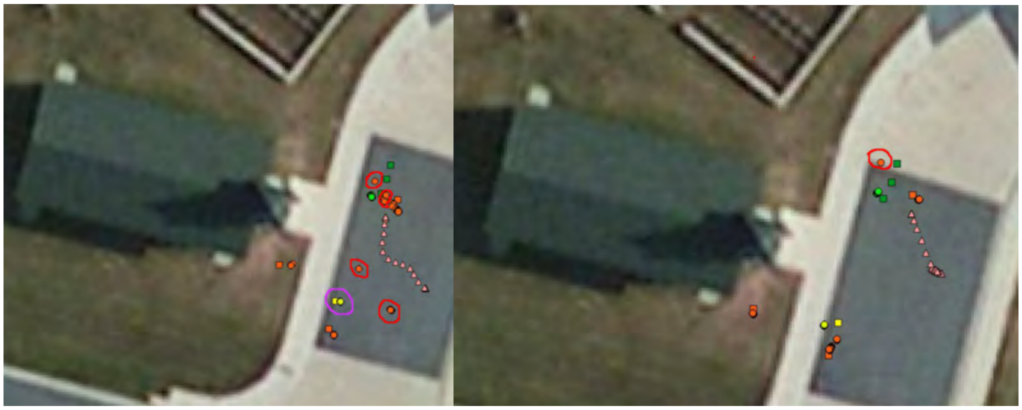


Fig. A-2 Run 5: left shows Husky 1, right shows Husky 3

Table A-3 Run 5 evaluation of perception

Husky	Alg	GT Obj.	FP Obj.	FN Obj.	False/All TB Obs.	False/All FH Obs.	False/All Car Obs.	False/All Gas Obs.
1	CMU	3 T, 1 F	4 T	0	1,476/2,577	0/343	NA	NA
1	UPenn	3 T, 1 F, 1 C	0	1 F	0/86	NA	0/80	NA
3	CMU	3 T, 1 F	1T	0	2/614	0/46	NA	NA
3	UPenn	3 T, 1 F, 1C	0	0	0/105	0/48	0/15	NA

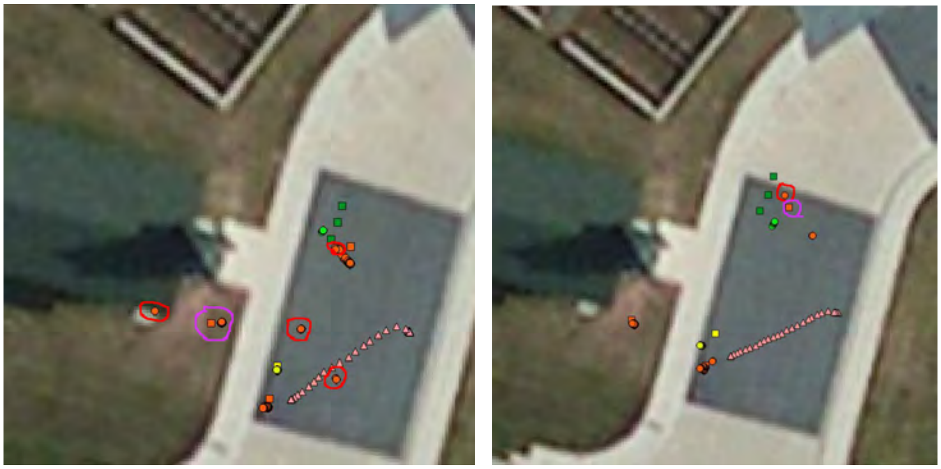


Fig. A-3 Run 6: left shows Husky 1, right shows Husky 3

Table A-4 Run 6 evaluation of perception

Husky	Alg	GT Obj.	FP Obj.	FN Obj.	False/All TB Obs.	False/All FH Obs.	False/All Car Obs.	False/All Gas Obs.
1	CMU	3T, 1 F	4 T	0	1,131/2,330	0/433	NA	NA
1	UPenn	3 T, 1 F, 1 C	0	1 T	0/128	0/8	0/34	NA
3	CMU	3 T, 1 F	1 T	0	3/383	0/32	NA	NA
3	UPenn	3 T, 1 F, 1C	0	1 T	0/67	0/26	0/9	NA

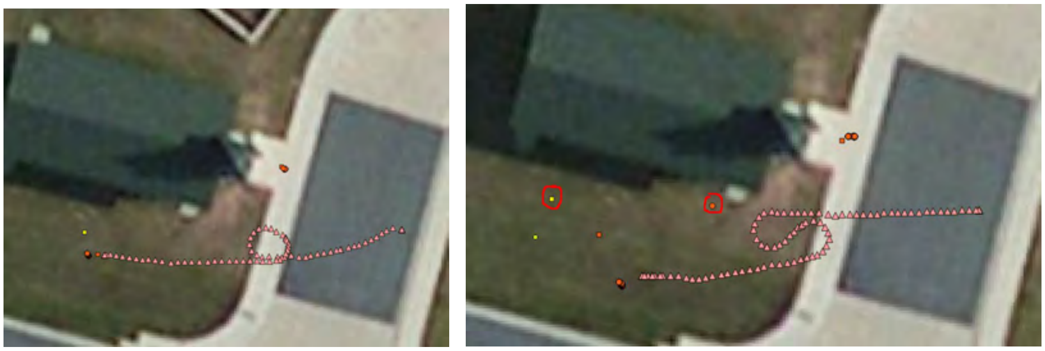


Fig. A-4 Run 7: left shows Husky 1, right shows Husky 3

Table A-5 Run 7 evaluation of perception

Husky	Alg	GT Obj.	FP Obj.	FN Obj.	False/All TB Obs.	False/All FH Obs.	False/All Car Obs.	False/All Gas Obs.
1	CMU	2 T, 1 F	0	1 F	0/291	NA	NA	NA
1	UPenn	2 T, 1F	0	1 F	0/8	NA	NA	NA
3	CMU	2 T, 1F	1 T	1 F	1/319	NA	NA	NA
3	UPenn	2 T, 1F	0	0	0/45	0/5	NA	NA

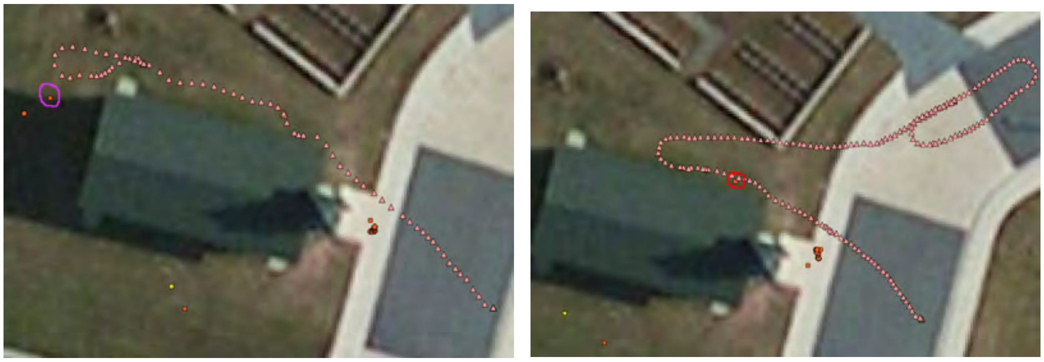


Fig. A-5 Run 8b/8: left shows Husky 1, right shows Husky 3

Table A-6 Run 8b/8 evaluation of perception

Husky	Alg	GT Obj.	FP Obj.	FN Obj.	False/All TB Obs.	False/All FH Obs.	False/All Car Obs.	False/All Gas Obs.
1	CMU	2 T	0	0	0/355	NA	NA	NA
1	UPenn	2 T	0	1 T	0/32	NA	NA	NA
3	CMU	2 T	1 T	0	2/289	NA	NA	NA
3	UPenn	2 T	0	0	0/35	NA	NA	NA

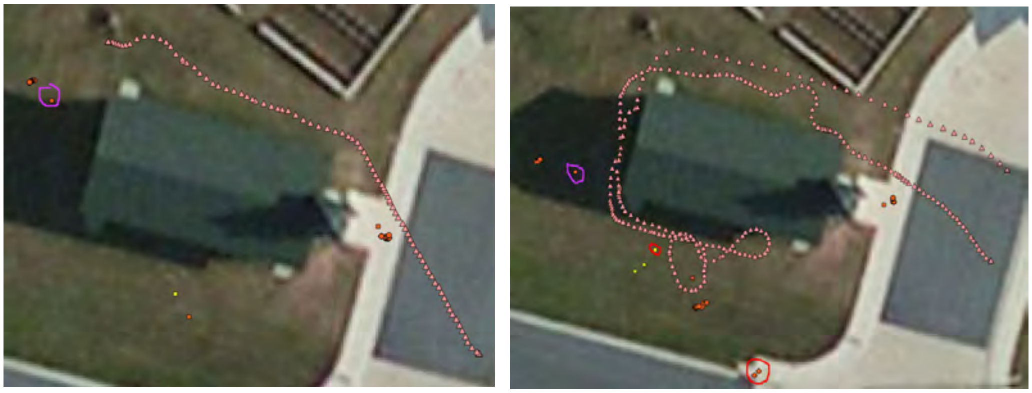


Fig. A-6 Run 9c/9: left shows Husky 1, right shows Husky 3

Table A-7 Run 9c/9 evaluation of perception

Husky	Alg	GT Obj.	FP Obj.	FN Obj.	False/All TB Obs.	False/All FH Obs.	False/All Car Obs.	False/All Gas Obs.
1	CMU	2 T	0	0	0/441	NA	NA	NA
1	UPenn	2 T	0	1 T	0/33	NA	NA	NA
3	CMU	3 T, 1F	1 T	1 F	163/998	0/57	NA	NA
3	UPenn	3 T, 1F	0	1 T	0/57	0/3	NA	NA

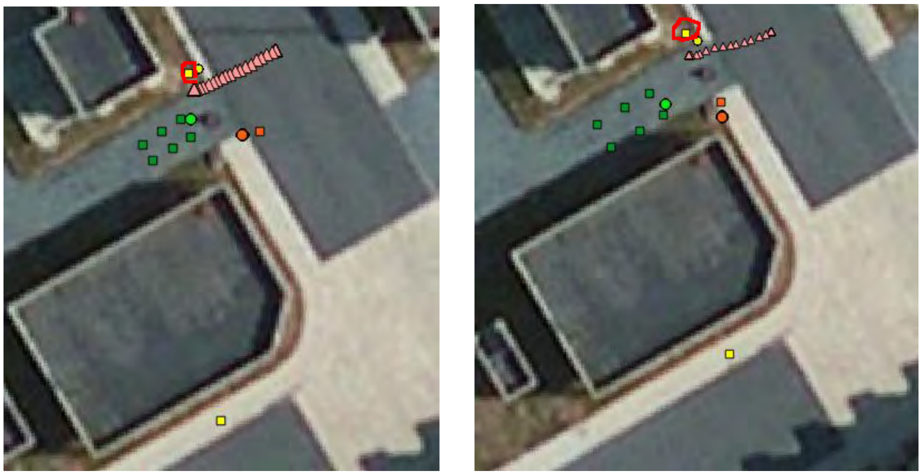


Fig. A-7 Run 21/21: left shows Husky 1, right shows Husky 3

Table A-8 Run 21/21 evaluation of perception

Husky	Alg	GT Obj.	FP Obj.	FN Obj.	False/All TB Obs.	False/All FH Obs.	False/All Car Obs.	False/All Gas Obs.
1	CMU	1 T, 1 F	0	1 F	0/232	0	NA	NA
1	UPenn	1 T, 1 F, 1 C	0	0	0/60	0/62	0/45	NA
3	CMU	1 T, 1 F	0	1 F	0/237	0	NA	NA
3	UPenn	1 T, 1 F, 1 C	0	0	0/73	0/69	0/67	NA

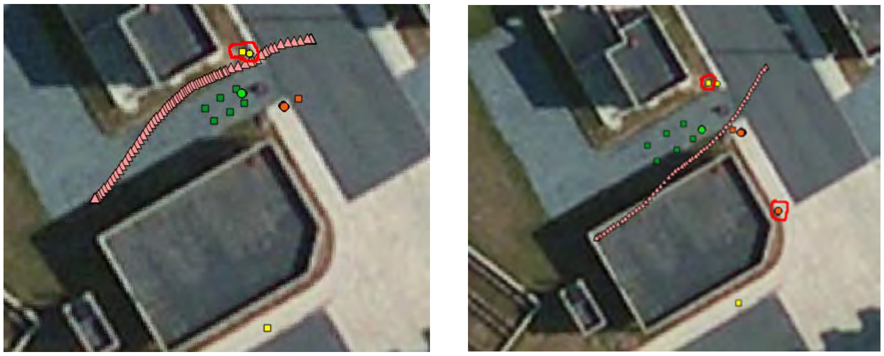


Fig. A-8 Run 22/22: left shows Husky 1, right shows Husky 3

Table A-9 Run 22/22 evaluation of perception

Husky	Alg	GT Obj.	FP Obj.	FN Obj.	False/All TB Obs.	False/All FH Obs.	False/All Car Obs.	False/All Gas Obs.
1	CMU	1 T, 1 F	0	1 F	0/285	0	NA	NA
1	Penn	1 T, 1 F, 1C	0	0	0/52	0/62	0/20	NA
3	CMU	1 T, 1 F	1 T	1 F	59/303	0	NA	NA
3	Penn	1 T, 1 F, 1C	0	0	0/68	0/2	0/31	NA

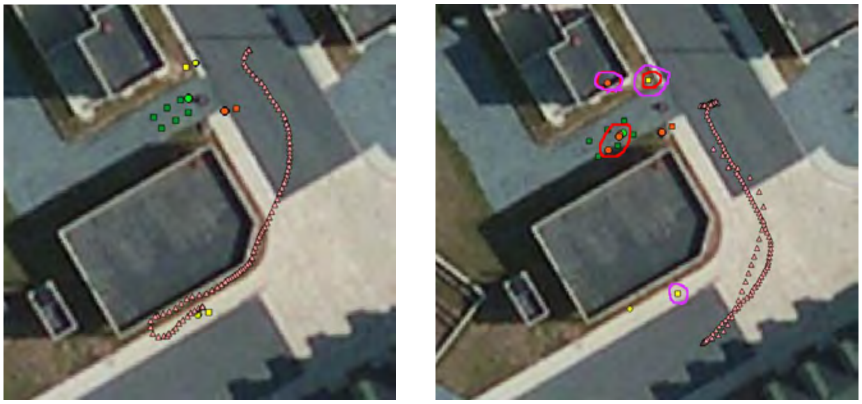


Fig. A-9 Run 23/23b: left shows Husky 1, right shows Husky 3

Table A-10 Run 23/23b evaluation of perception

Husky	Alg	GT Obj.	FP Obj.	FN Obj.	False/All TB Obs.	False/All FH Obs.	False/All Car Obs.	False/All Gas Obs.
1	CMU	1 T, 1 F	0	0	0/466	0/80	NA	NA
1	UPenn	1 T, 1 F, 1 C	0	0	0/66	0/68	0/30	NA
3	CMU	1 T, 2 F	1 T	1 F	91/472	0/1	NA	NA
3	UPenn	1 T, 2 F, 1 C	1 T	2 F	1/97	0	0/2	NA

In the replication of run 24 on Husky 3, 2 traffic barrels were placed on the left side of the gas pumps (left from the reader's point of view), with one being moved to the right side before the start of the run. Thus, there were 3 traffic barrels observable but only 2 used during the run as described in Section 3, Table 2.

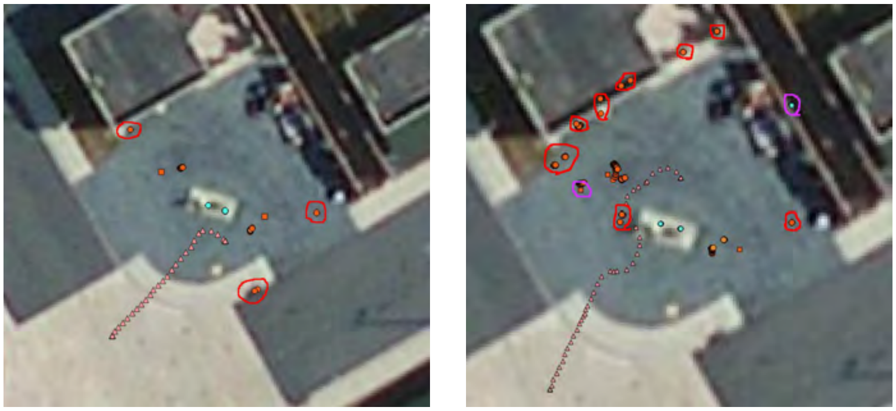


Fig. A-10 Run 24/24: left shows Husky 1, right shows Husky 3

Table A-11 Run 24/24 evaluation of perception

Husky	Alg	GT Obj.	FP Obj.	FN Obj.	False/All TB Obs.	False/All FH Obs.	False/All Car Obs.	False/All Gas Obs.
1	CMU	2 T	3 T	0	97/874	NA	NA	NA
1	UPenn	2 T, 2G	0	0	0/26	NA	NA	0/299
3	CMU	3 T	7 T	0	231/991	NA	NA	NA
3	UPenn	3 T, 2G	1 G	1 T	0/16	NA	NA	24/355

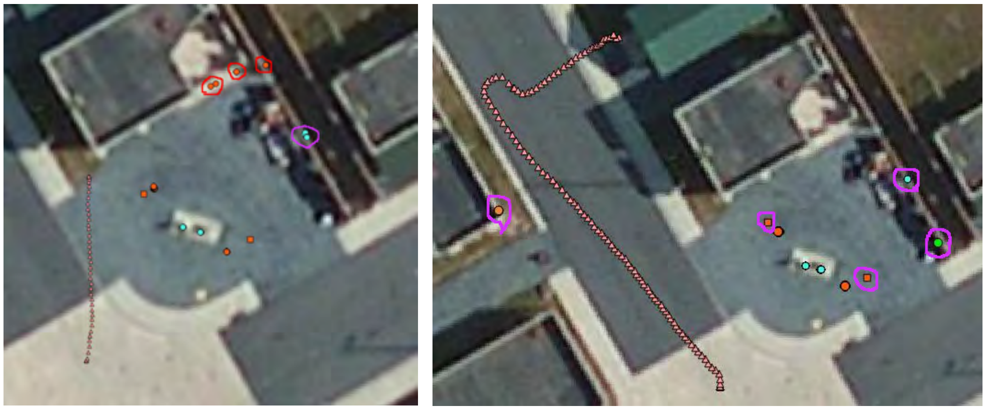


Fig. A-11 Run 25b/25: left shows Husky 1, right shows Husky 3

Table A-12 Run 25b/25 evaluation of perception

Husky	Alg	GT Obj.	FP Obj.	FN Obj.	False/All TB Obs.	False/All FH Obs.	False/All Car Obs.	False/All Gas Obs.
1	CMU	2 T	3 T	0	57/529	NA	NA	NA
1	UPenn	2 T, 2G	1 G	0	0/33	NA	NA	7/113
3	CMU	2 T	0	0	0/772	NA	NA	NA
3	UPenn	2 T, 2G	1T, 1G, 1C	2T	0/6	NA	1/1	4/228

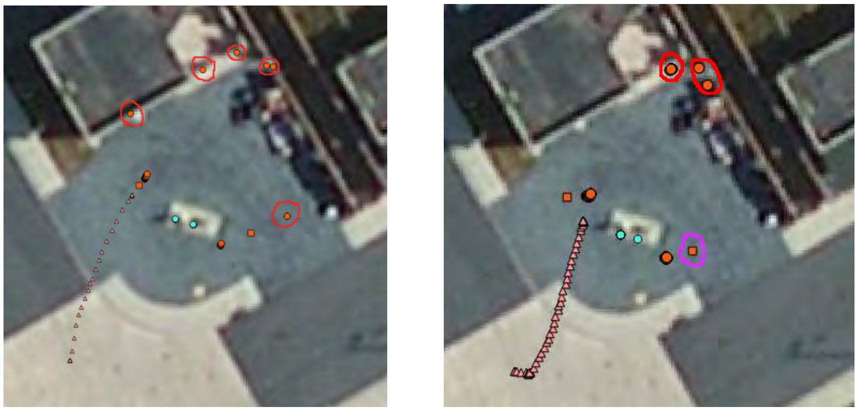


Fig. A-12 Run 26/26: left shows Husky 1, right shows Husky 3

Table A-13 Run 26/26 evaluation of perception

Husky	Alg	GT Obj.	FP Obj.	FN Obj.	False/All TB Obs.	False/All FH Obs.	False/All Car Obs.	False/All Gas Obs.
1	CMU	2 T	5 T	0	49/341	NA	NA	NA
1	UPenn	2 T, 2G	0	0	0/36	NA	NA	0/69
3	CMU	2 T	2 T	0	248/942	NA	NA	NA
3	UPenn	2 T, 2G	0	1 T	0/6	NA	NA	0/109

INTENTIONALLY LEFT BLANK.

List of Symbols, Abbreviations, and Acronyms

2-D	2-dimensional
3-D	3-dimensional
ADPM	active deformable part model
CACTF	Combined Arms Collective Training Facility
CMU	Carnegie Mellon University
CWM	common world model
DPM	deformable part model
HRI	human-robot interaction
RANSAC	RANdom SAmple Consensus
RCTA	Robotics Collaborative Technology Alliance
TBS	tactical behavior specification
UPenn	University of Pennsylvania

1 DEFENSE TECHNICAL
(PDF) INFORMATION CTR
DTIC OCA

2 DIRECTOR
(PDF) US ARMY RESEARCH LAB
RDRL CIO LL
IMAL HRA MAIL & RECORDS
MGMT

1 GOVT PRINTG OFC
(PDF) A MALHOTRA

1 DIRECTOR
(PDF) US ARMY RESEARCH LAB
RDRL VTA
C LENNON